REMARKS

STATUS OF THE CLAIMS

Claims 53-55 and 58 are pending in the application.

Claims 53-55 and 58 are rejected under 35 USC 102(b) as being anticipated by Nelson, "Running Visual Basic for Windows," Microsoft Press, 1993 (hereinafter referred to as "Visual Basic"). Visual Basic is newly cited, and, thus, newly relied upon.

Thus, claims 53-55 and 58 remain pending for reconsideration, which is respectfully requested.

REJECTION

The remarks herein further clarify the previous Amendment arguments, and it is believed that the claimed present invention as recited in independent claims 53, 55 and 58 are allowable over Visual Basic.

The Office Action rejection newly relies on Visual Basic, which is an object oriented programming language developed by MICROSOFT based upon the BASIC programming language. In particular, visual basic is a visual programming environment used for developing WINDOWS applications. In Visual Basic, a programmer can design graphical user interface (GUI) applications graphically by dragging program elements (as objects), represented by icons, from the Visual BASIC Toolbox, and can write BASIC code for each program element. Visual BASIC is event-driven; events can be, for example, a procedure is called automatically when the end user chooses menu items, clicks the mouse, moves objects on the screen, etc.

Therefore, the Office Action appears to rely on the Visual Basic programming environment, which allows designing a GUI application by graphically creating components (objects) of the application and defining attributes of the objects, which can include executable code associated with an object (pages 5 and 12-13 of Visual Basic). In other words, the Office Action rejection on page 3 of the Office Action alleges that the GUI application objects created in visual basic would be similar to the claimed present invention's object oriented programming (OOP) object recited, for example, in independent claim 53.

4

However, in contrast to a Visual Basic object, the claimed present invention as recited in independent claims 53, 55, and 58, using claim 53 as an example, provides:

> 53. (PREVIOUSLY PRESENTED) A computer readable storage that stores object oriented programming objects accessible by objects, comprising:
>
> a data structure that defines *a component to serve as one object* in combination with existing software having a graphical user interface, *by storing data related to the existing software having the graphical user interface and storing as an operation method related to the existing software, a program code controlling a data processing system according to a process of issuing an event of the existing software in response to a received message issued in another object* (emphasis added).

More particularly, in contrast to a typical Visual Basic object defined in the Visual Basic programming environment, the claimed present invention, as recited in independent claims 53, 55 and 58, is directed to defining an object to work in combination with and/or to replace an existing software module that has a graphical user interface (i.e., to work with and/or to replace an existing GUI application, with another object), as discussed in the paragraph spanning pages 175-176 and pages 176-177 of the present Application. In contrast to a typical Visual Basic object, the claimed present invention's object, as shown in FIG. 101, provides a component A (an object) "*including*" an "*existing software having a graphical user interface*" (claim 53) (page 178, lines 4-19 of the present Application, emphasis added).

Typically, an OOP object has a DATA portion and an associated operation METHOD portion, as shown in FIG. 101 and discussed in Visual Basic pages 12-13. The concept of "*including*" in the claimed present invention, as shown in FIG. 101, relates to defining in an object (can be called a software replacement object or component A object), a DATA portion that includes data related to an existing GUI software that is (in this example) being replaced (e.g., in FIG. 101, DATA portion: Application (A.exe) and window information is the object DATA portion that has data related to the existing GUI Application A that is being replaced). The concept of "*including*" in the claimed present invention, as shown in FIG. 101, also relates to defining in such software replacement object (component A) an operation METHOD portion that is an existing application drive program to drive the existing application being replaced (in FIG. 101, the GUI Application A is being replaced and the METHOD portion of Component A is the existing application drive program.

With reference to FIG. 101, a benefit of the claimed present invention is to replace an existing GUI application A by an object by simply adding the Component A and maintaining the existing GUI application A as it is (page 179 of the present Application). More particularly, a benefit of the claimed present invention is that upon receipt of an issue of an event of an existing software A, to execute on a cooperative basis a method which does not appear on a specification of the existing software A, without adding advanced functions to the existing software A (page 181, lines 4-8 and FIG. 103).

More particularly, the Office Action on page 3 of the Office Action relies on Visual Basic's object properties (FIG. 1-2), to reject the claimed present invention's, "a data structure that defines *a component to serve as one object* in combination with existing software having a graphical user interface, *by storing data related to the existing software having the graphical user interface*." However, object properties of Visual Basic, as discussed in page 6 of Visual Basic, describe attributes of an object itself, which differ from the claimed present invention's object "*storing data related to the existing software having the graphical user interface*." In other words, none of the object property attributes discussed in Visual Basic (e.g., FIG. 1-5) relate to claimed present invention's *object DATA portion, which stores "the data related the existing software having the graphical user interface*."

Further, the Office Action on page 3 of the Office Action relies on page 151-163 of Visual Basic, concerning passing parameters among objects, to reject the claimed present invention's, "a data structure that defines *a component to serve as one object* in combination with existing software having a graphical user interface, by ... *storing as an operation method related to the existing software, a program code controlling a data processing system according to a process of issuing an event of the existing software in response to a received message issued in another object*." However, Visual Basic's discussion of user defined callable routines associated with an object, on pages 151-163 of Visual Basic, clearly differs from the claimed present invention's *object METHOD portion, which stores "a program code controlling a data processing system according to a process of issuing an event of the existing software in response to a received message issued in another object*." In particular, page 154 of Visual Basic expressly discloses that Visual Basic "does not allow user-defined methods."

Therefore, in view of the remarks, the claimed present invention as recited in independent claims 53, 55 and 58 is patentably distinguishing over Visual Basic, because Visual Basic fails to disclose every aspect (element) of the claimed present invention, either expressly,

impliedly or inherently, so that Visual Basic cannot anticipate the claimed present invention. Withdrawal of the anticipatory rejection of pending claims and allowance of pending claims is respectfully requested.
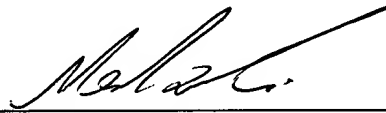
CONCLUSION

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

Respectfully submitted,
STAAS & HALSEY LLP

Date: _____June 22, 2005_____   By: _____

Mehdi D. Sheikerz
Registration No. 41,307

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501